X-692-71-200
PREPRINT

NASA TM X- 65618

SPLINE QUADRATURE

J. D. SCUDDER

JULY 1971

GSFC — GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-692-71-200

SPLINE QUADRATURE

by

J. D. Scudder
Laboratory for Extraterrestrial Physics
NASA Goddard Space Flight Center
Greenbelt, Maryland

July 1971

ABSTRACT

A non-iterative quadrature algorithm is presented which is based on a cubical spline interpolant. Since the underlying interpolant can even interpolate data which is unequally spaced in abscissa, the proposed quadrature algorithm can equally handle this type of data or equispaced data. Since the underlying interpolant is the smoothest C^2 function which interpolates all the data, numerical quadrature by this technique minimizes spurious contributions to the integral which often result by least squares techniques, Simpson's 1/3 or other similar multiple rules which implicitly employ a polynomial interpolate. A direct comparison for equispaced data between this algorithm, Simpson's 1/3 rule, and the exact analytic quadrature for a wide range of polynomial and transcendental integrands shows that the algorithm is generally five times more accurate than Simpson's 1/3 rule.

The algorithm requires as input information i) the data to be integrated and ii) some approximation to the leading and trailing slopes of the interpolant. For the cases of analytical integrands condition ii) is easily met by the elementary calculus. For tabular data, the loosely coupled nature of the underlying interpolant is shown to manifest itself by confining inaccurate leading and trailing slopes effects on the interpolant to at

worst the first and last three data points. By sampling outside the interval of interest one can then obtain more accurate, smooth interpolation and therefore quadrature on the inner interval of the interpolant's domain.

INTRODUCTION

Determining integrals of tabulated empirical, or otherwise discretely known data is a general numerical problem which is often encountered in scientific work. This task is often complicated by the constraints that result from data sampling. Rarely is one fortunate in having the data equally spaced, or, if one is so fortunate, the number of data points is generally not some exact multiple of specified numbers, N, peculiar to the ordinary multiple rules such as those of Simpson, Hardy, or Weddle. Even in the case when the data is equispaced and such a multiple of N, the underlying interpolation of the discrete data is by a continuous function which is a piecewise polynomial of order N-1, just as one would obtain by Lagrange interpolation over N points. The pitfalls of Lagrange interpolation are well known and are generally to be avoided in numerical work, especially as N becomes large. The basic problem with Lagrange interpolation is that it oscillates through the domain of interpolation in order to interpolate every data point.+

If the underlying objective in taking discrete data and interpolating the results is to approximate a continuous function,

+For graphic examples cf. Thompson p. 18.

Lagrange interpolants may introduce much spurious content. The integral of such a poor approximation of the continuous function may therefore be a poor approximation to the integral of the underlying continuous function. Least squaring the discrete approximations of the integrand to some functional form is also used for numerical quadratures of equally or unequally spaced data. This technique is not useful if one is trying to infer from the integration something about the underlying functional form of the data. This is often the purpose in taking moments of empirical distributions.

An alternative quadrature technique which reduces the problems of injecting spurious information in the interpolation step prior to quadrature will now be discussed. Often it is an implicit or explicit assumption in sampling with discrete counters (generic) that the underlying structure of "events" evolves smoothly in time, increasing energy or other variable.+ An example (explicit) of such a "counter" would be the discrete monotonic tabulation of a continuously differentiable function at a number of points (= "events") in its domain. Another example (implicit) would be a series of geiger counters with different energy thresholds monitoring the distribution of energetic solar electrons. The assumption is rather common. When discrete data are taken under this assumption(and all instrumental noises

+This assumption will be referred to as the smooth distribution hypothesis

are removed) the smoothest curve which interpolates the data is the only interpolant which does not interject further assumptions about the underlying distribution of "events". Theoretical cubical splines are in the class of such interpolants.

Recently cubical spline interpolative techniques have been reduced to non-iterative algorithmic forms of exceptional computational stability and accuracy suitable for digital computers (Thompson 1970, 1971). The resulting interpolant S(x) interpolates every data point regardless of spacing in such a manner that S(x) possesses a maximal smoothness property as compared with <u>any</u> other C^2 function which interpolates all the data, (Holladay, 1957). By construction, the interpolant on any interval between data points "knows" about the variation of the data in the neighboring intervals so that it may make the "smoothest" transition from that interval to the next and still interpolate <u>all</u> the subsequent data. This contrasts sharply with the local nature of the interpolants of the <u>multiple rules</u> whose derivatives are discontinuous at data points indexed jN, $\{j = 0,1,2,\ldots\}$.

The present quadrature algorithm exploits the piecewise (in general different) cubic functional form of S(x) on each subinterval between data points to do the necessary integrals via Simpson's 1/3 rule which is exact for polynomials of order

≤ 3. Since the spline interpolant considered over the entire interval of interest is one of maximal "smoothness", sharp junctures in the interpolant are avoided.

GENERAL SCHEME

The integration technique has been formulated as a computer subroutine, INTEG, which computes numerical integrals for tabulated data $\{(x_i, U(x_i), i = 1, NPTS = \text{number of data points}\}$ equally or unequally spaced in abscissa. This modular routine is designed to be compatible and interface with and use the routines developed for spline interpolation by Thompson (1970). The spline function S(x) is fitted to the data in the non-iterative method of Thompson (1970) using the computer subroutine SPLN2.[+]

S(x) is continuous, and therefore

$$\int_{x_1}^{x_{NPTS}} U(x)dx \simeq \int_a^b S(x)dx = \sum_{i=1}^{NPTS-1} \int_{x_i}^{x_{i+1}} S(x)dx, \tag{1}$$

where U(x) is the underlying (smooth) distribution from which samples are taken. Employing the interpolative power which knowledge of S(x) implies, and using the fact that S(x) restricted

[+]Copies of this and other routines referenced in Thompson 1970 may be found in Appendix B.

to $[x_i, x_{i+1}]$ is a cubic polynomial for which Simpson's 1/3 rule is exact, we obtain:

$$\int_{x_i}^{x_{i+1}} S(x)dx = \frac{\delta_i}{3}(S(x_i) + 4S(x_i + \delta_i) + S(x_{i+1})) \tag{2}$$

where

$$\delta_i = \frac{x_{i+1} - x_i}{2} .$$

Since $S(x_i) \equiv U(x_i)$

$$\int_a^b U(x)dx \simeq \frac{1}{3} \sum_{i=1}^{NPTS-1} \delta_i(U(x_i) + 4S(x_i + \delta_i) + U(x_{i+1})) \tag{3}$$

COMPARISON OF SIMPSON AND SPLINE QUADRATURE

It is worthwhile to inquire whether this technique has any computational advantages over the equispaced rules. To demonstrate the advantage of spline quadrature we consider the following class of numerical problems: integrals without antederivatives. Although these integrands possess no antederivatives they generally possess analytic expressions for the derivative via the elementary calculus. As we shall see this will be important. In general to get the best interpolant one should provide good leading and trailing slopes (Q1,QN) for $S(x)$.

In the comparison of spline and Simpson quadrature we place the two techniques on equal a priori footing. This means (i)

that the number of data points NPTS must be odd so that Simpson's rule may be used; INTEG, however, only requires NPTS > 3, therefore let NPTS = {5,7,9,11,....} for the comparison; and (ii) that we compute analytically the derivative of the integrand so that the spline interpolant is not biased. Later we will discuss the problem of leading and trailing slopes for discrete data. In order to appraise the relative merits of these two techniques in the case where analytical methods fail, we will compare them for functions which do have antederivatives. These examples provide an absolute reference for the quadrature determination. Since neither condition is violated for integrals without antederivatives the results of these comparisons should then follow directly for them.

In fig. 1 are plotted the percentage absolute error for Simpson's 1/3 and spline quadrature for three cases. Additional examples are tabulated in Appendix A for a wider range of functions including other transcendentals. It can be seen that spline quadrature in all examples has a smaller absolute error than does Simpson's 1/3 for a rather diverse range of functions. As one departs from polynomials of order ≤ 3 for which spline and Simpson agree to 9 decimal degits and for which Simpson is exact, we see that spline quadrature is always

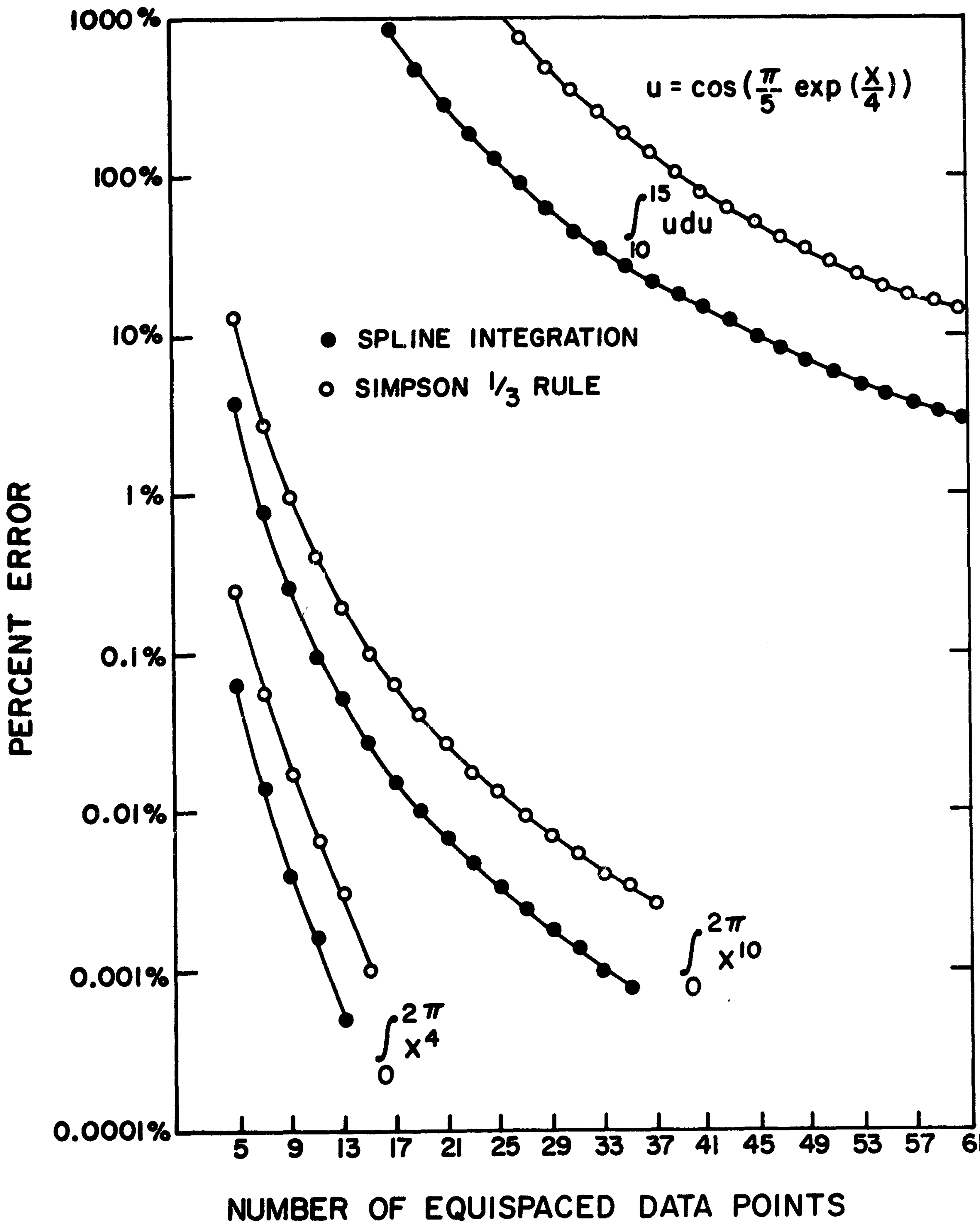

Figure 1. Graphical comparison of the errors Simpson 1/3 and spline Quadrature for integrals indicated as a function of equispaced data points defining the integrand.

$\sim$ 5 times less error prone for the same number of data points than Simpson. As the integrands develop more variability the absolute advantage becomes decidedly appreciable. The $\int udu$ in the upper right hand corner was "constructed" to illustrate this fact. The integrand is shown in fig. 2. The arguments are driven non-linearly which gives rise to the aperiodicity so that fortuitous equispacing of data samples could not give an a prior advantage to Simpson. The absolute error after 61 equispaced data points is 2.79% vs. 12.65% for spline and Simpson respectively. This is $\sim$ 10% better absolute sensitivity of spline quadrature versus Simpson. Though the absolute size of the error may vary with the interval, the 5 times more accurate statement is also seen to hold in this example. This greater flexibility of the quadrature is a direct manifestation of the global smoothness of the interpolant in contrast to the jointed polynomial interpolations which are the foundation of the multiple quadrature rules.

Since spline quadrature is more accurate for a fixed tabulation of a known function whose antederivative is known, the same relative merit of spline vs Simpson quadrature is suggested for integrands whose antederivatives are not known, since the analytical existence of the antederivative in the comparison above was not used.

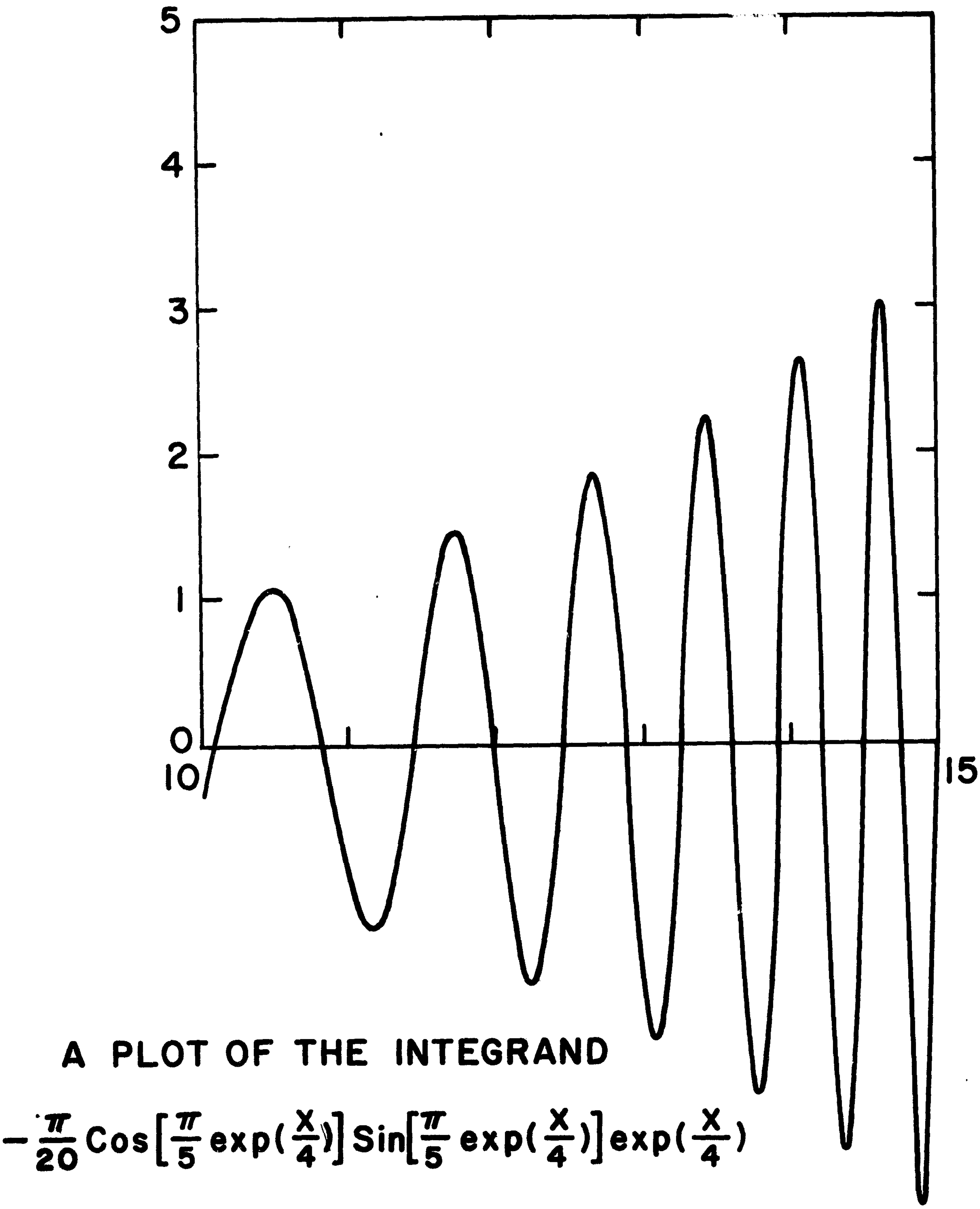

Figure 2. A graph of the integrand udu which is asynchronously driven and used in the comparison in figure 1.

GENERAL QUADRATURES

For discrete sampling from an unknown distribution we seemingly have replaced the problem of numerical quadrature with that of determining the derivatives of the unknown distribution function so as to provide the initial and trailing slopes for the interpolant S(x). Numerical derivatives are a source of pitfalls by themselves. The current interpolative scheme of Thompson is computationaly speaking, loosely connected, i.e. the sparse matrix involved in the linear solution for the spline is mostly zeroes. (Thompson, 1970 p. 5) The value of the leading and trailing slopes (Q1 and QN below) will propagate through the resulting solution for the smoothest curve which interpolates all the data with these values as boundary conditions. The loosely connected structure of the algorithum implies, however, that the effects of a poor initial slope for a given set of data will not cause ringing more than two to three intervals away from the end point . To illustrate the extent to which bad leading and trailing slopes can affect the interpolant we now consider figures 3 and 4. In figure 3 we have plotted isometrically X = abscissa for the function y =•4 X + 5. Y in the figure is the percentage error of the spline interpolant vs y at each X, when QN = -4 and Q1 = Z. Plotted here are a series of such error plots for Z in the range [-14,6] in two unit intervals.

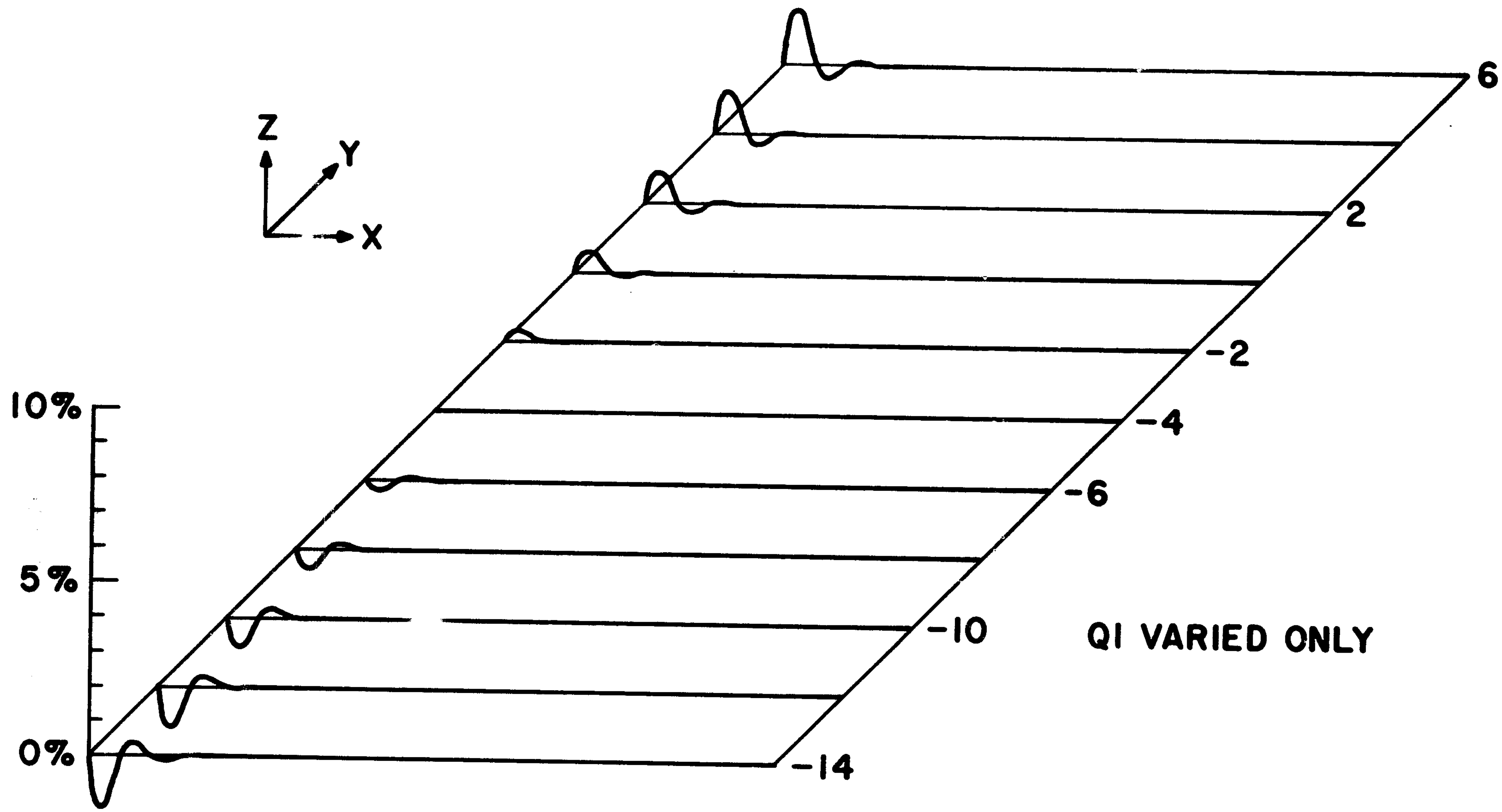

Figure 3. Consult text page 12

Since the spline interpolates <u>every</u> data point, the nodes in Y correspond to the x positions of the data points. The trailing slope in example three was obtained from the analytic form of y(x) and is exact. The leading slope, Q1, is varied from extreme to extreme to show that <u>very</u> poor approximations to the leading slope of the line propagate at most 3 data points away even with 10° errors in the leading slope. Figure 4 shows a similar result when both Q1 and QN are simultaneously in error. The same loosely coupled result is clearly shown.

In practice the linear approximation to the data will not be a disastorous mistake or be the cause of serious ringing. Making sure that a reasonable slope, i.e. one not inconsistent with the trend of the data, is a small price to pay for a technique otherwise free of spurious content. An example of such a strategy is shown in figure 5. The X and Z axes are the same as in figures 3 and 4, whereas the Y axis is now the number of equally spaced data points used in approximating the distribution sampled from $y = \sin x + 50$. The leading and trailing slopes used for the interpolant were linear approximations determined from the data. Note the changes in scale from figures 3 and 4. The errors under such a scheme are not very substantial.

For discrete data a linear approximation to the data is a way of injecting minimal distortion into the interpolant. The

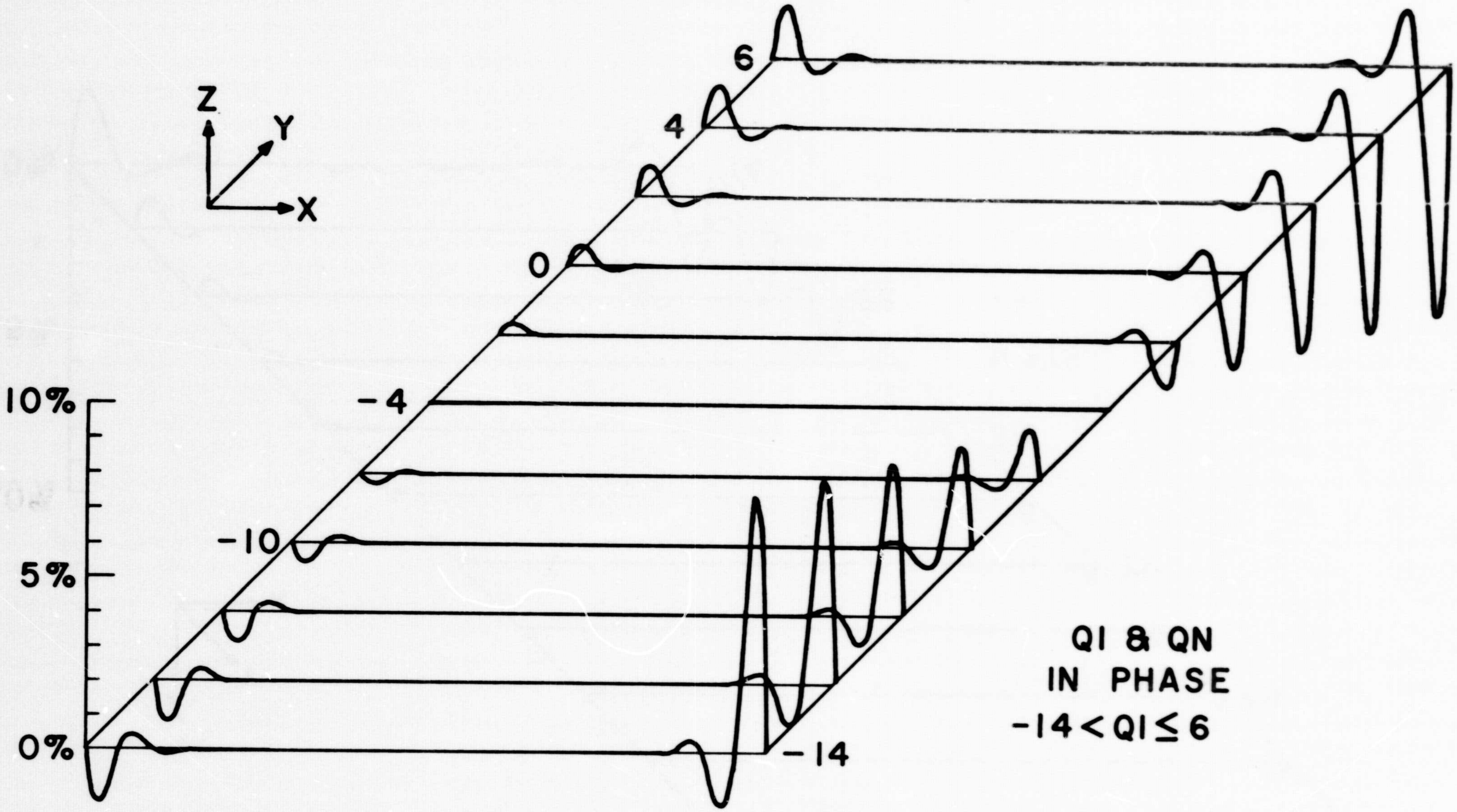

Figure 4. Consult text page 14

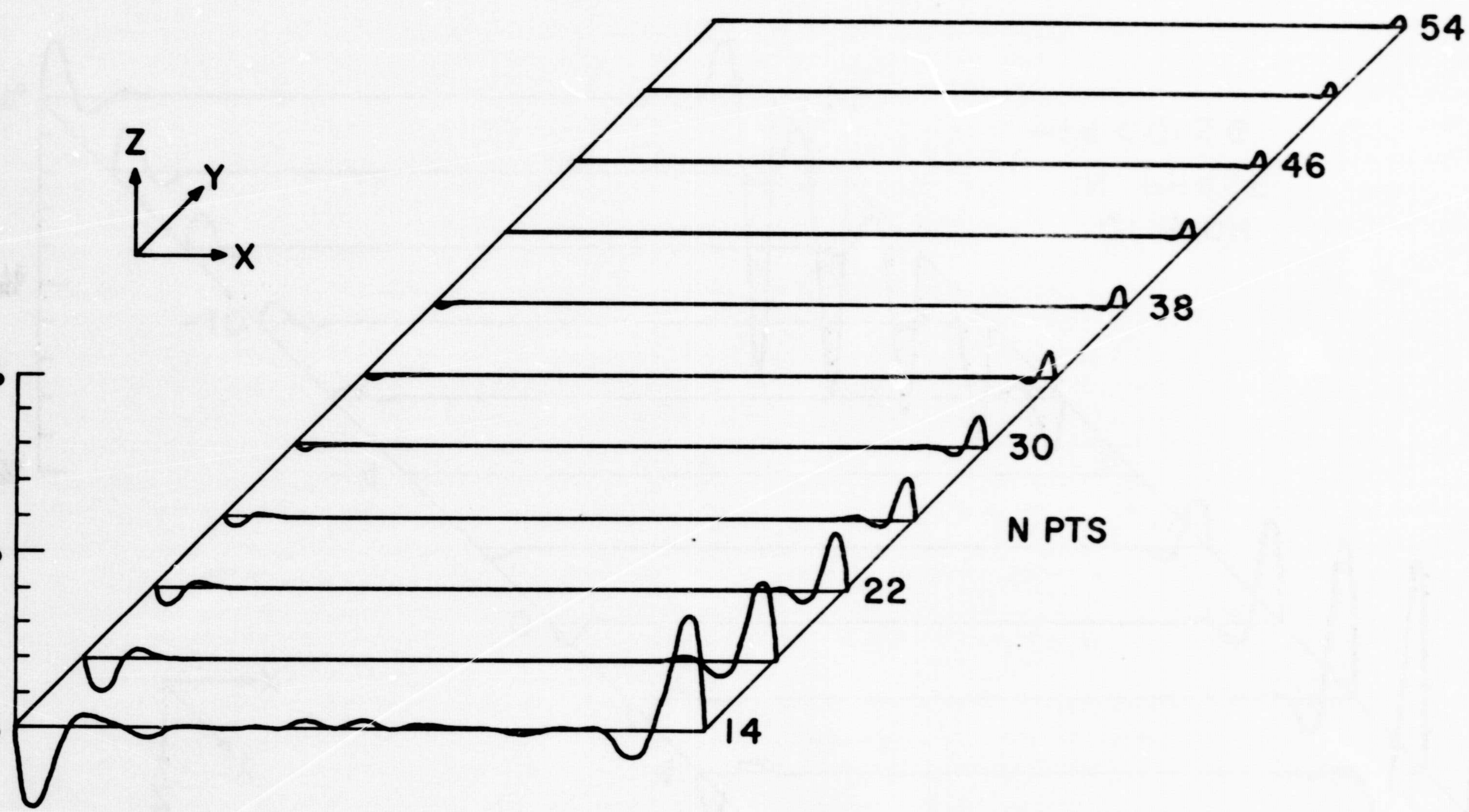

Figure 5. Consult text page 14

loosely coupled nature of the interpolant implies that an important strategy for good interpolants over some specified experimental domain would be to sample outside this domain to allow the interpolant to settle down from the end effects which result from the slope approximation.

It is thus the loosely coupled nature of the interpolant which allows one to say that the success of this numerical quadrature rule has not been undermined by the "hidden slopes" problem of the underlying spline interpolant.

FLEXIBILITY OF THE ALGORITHM: A Significant Advantage

All of the above discussion has been from examples where the data is equispaced. The underlying interpolant is not restricted to data equispaced in abscissa. This adaptability of the spline interpolant makes the quadrature algorithm similarly flexible, for nothing in the algorithm depends on the spacing of the discrete data. Therefore in this algorithm is found the best quadrature determination regardless of data spacing consistent with the smooth distribution hypothesis. It should not be misconstrued from this last statement that this algorithm can be used blindly on any data set. One should make sure that the smooth distribution assumption is justifiable, i.e. interpolation is meaningful, before relying on the accuracy of this algorithm. Wildly fluctuating ordinates coupled with the global smoothness criterion for the underlying interpolant will often make the

quadrature result by this technique meaningless. This algorithm is built on an interpolation algorithm. If you would trust the results of the smoothest C^2 interpolant you can trust the results of the quadrature algorithm.

IMPLEMENTATION

The main and auxiliary routines were written for use on the IBM 1800. For usage with more sophisticated compilers, however, there exist some obvious optimizations which will not be considered here.

```
SUBROUTINE INTEG(SUM,ISLOP,IPRNT)
```

1). The data points of the function to be integrated, (X,U(X)), should be transmitted through COMMON as should NPTS, the number of data points to be interpolated and Q1 and QN, the initial and final slopes of the integrand.

```
COMMON X(200),U(200),S(200),DEL(200),Q1,QN,NPTS
```

2). If the data is tabular, decide whether linear approximations to the solpes at X_1 and X_{NPTS} are adequate. They are called Q1 and QN respectively. More elaborate empirical slopes can be devised and loaded as subroutine D(X) if desired. If the integrand is analytical it is desirable to load for D(X) the analytical function U'(X) = D(X).

```
CALLING PARAMETERS

     ISLOP = 1- INTEG WILL COMPUTE A LINEAR APPROXIMATION
     TO Q1 AND QN,THE INITIAL AND FINAL SLOPES OF THE INTERPOLANT
     ISLOP = 2- INTEG WILL LOOK FOR AND USE A FUNCTION SUBROUTINE
     D(X) FOR SOME OTHER APPROXIMATION OR EXPLICIT ANALYTICAL
     FORM OF THE DERIVATIVE OF THE INTEGRAND IN EVALUATION OF
     Q1 AND QN.
```

```
N.B. IN EITHER CASE SOME FUNCTION SUBROUTINE D(X) MUST BE
COMPILED WITH INTEG WHETHER DUMMY OR REAL. IT IS MOST
IMPORTANT THAT ONE SUPPLY THE BEST KNOWN VALUES FOR D1 AND DN.
FOR FURTHER DETAILS CONCERNING THE SPLINE, CF. NASA-GSFC
X-692-70-261,'SPLINE INTERPOLATION ON A DIGITAL COMPUTER'
BY R.F.THOMPSON OR 'SPLINE QUADRATURE' X-692-71-200 BY
J.D.SCUDDER
```

3). If matters of convergence of an integral are important, or if one desires to see from which interval comes most of the integral's value a simple adjustment of the IPRNT parameter will dump the areas bounded by $[X_i, X_{i+1}]$ with the tag i+.5 to the left.

```
IPRNT = 1 INTEG WILL
                         WRITE OUT THE PANEL SUMS
                         CONTRIBUTING TO THE INTEGRAL
IPRNT = 2 INTEG WILL NOT
```

4).

```
SUBROUTINES CALLED

     SPLN2 WHICH FILLS ARRAYS S AND DEL AND MUST BE CALLED
     BEFORE SPLIT(J) IS USED.
```

5).

```
FUNCTIONS EMPLOYED

     SPLIT(J) WHICH RETURNS THE INTERPOLATED VALUE FOR
     S(X) AT THE MIDPOINT OF THE J'TH INTERVAL
     BETWEEN DATA POINTS.

     D(X) WHICH COMPUTES A USER SUPPLIED APPROXIMATION TO THE
     DERIVATIVE IF BETTER APPROXIMATION THAN THE LINEAR ONE
     SUPPLIED IS DESIRED.
```

***An example of such a dummy function subprogram would be**

```
FUNCTION D(X)
D = 0.
RETURN
END
```

```
      SUBROUTINE INTEG(SUM,ISLOP,IPRNT)
      COMMON X(100),U(100),S(100),DEL(100),Q1,QN,NPTS
            INTEG = INTEGRATE IS AN ALL PURPOSE NUMERICAL INTEGRATION
            ALGORITHM.  THE FUNCTIONAL PAIRS X(I),U(I) ARE COMMUNICATED
            THROUGH COMMON AS IS NPTS = NUMBER OF DATA POINTS GREATER THAN 3.
            THESE VARIABLES AND ARRAYS MUST BE DEFINED IN THE CALLING
            PROGRAM.   THE ABSCISSAE,X(I),NEED ONLY BE DISTINCT AND MONO-
            TONIC INCREASING. N.B. THIS ALLOWS NUMERICAL INTEGRATION OF
            IRREGULARLY SPACED OR TABULATED DATA.
            THE RETURNED VALUE SUM IS EQUAL TO THE DEFINITE INTEGRAL OF
            S(X)DX OVER THE RANGE X(1),X(NPTS).

            CALLING PARAMETERS

                  ISLOP = 1- INTEG WILL COMPUTE A LINEAR APPROXIMATION
                  TO Q1 AND QN,THE INITIAL AND FINAL SLOPES OF THE INTERPOLANT
                  ISLOP = 2- INTEG WILL LOOK FOR AND USE A FUNCTION SUBROUTINE
                  D(X) FOR SOME OTHER APPROXIMATION OR EXPLICIT ANALYTICAL
                  FORM OF THE DERIVATIVE OF THE INTEGRAND IN EVALUATION OF
                  Q1 AND QN.
                  N.B. IN EITHER CASE SOME FUNCTION SUBROUTINE D(X) MUST BE
                  COMPILED WITH INTEG WHETHER DUMMY+OR REAL. IT IS MOST
                  IMPORTANT THAT ONE SUPPLY THE BEST KNOWN VALUES FOR Q1 AND QN.
                  FOR FURTHER DETAILS CONCERNING THE SPLINE, CF. NASA-GSFC
                  X-692-70-261,'SPLINE INTERPOLATION ON A DIGITAL COMPUTER'
                  BY R.F.THOMPSON OR 'SPLINE QUADRATURE' X-692-71-200 BY
                  J.D.SCUDDER

                  IPRNT = 1 INTEG WILL
                                             WRITE OUT THE PANEL SUMS
                                             CONTRIBUTING TO THE INTEGRAL
                  IPRNT = 2 INTEG WILL NOT

            SUBROUTINES CALLED

                  SPLN2*WHICH FILLS ARRAYS S AND DEL AND MUST BE CALLED
                  BEFORE SPLIT(J)*IS USED.

            FUNCTIONS EMPLOYED

                  SPLIT(J)*WHICH RETURNS THE INTERPOLATED VALUE FOR
                  S(X) AT THE MIDPOINT OF THE J'TH INTERVAL
                  BETWEEN DATA POINTS.

                  D(X) WHICH COMPUTES A USER SUPPLIED APPROXIMATION TO THE
                  DERIVATIVE IF BETTER APPROXIMATION THAN THE LINEAR ONE
                  SUPPLIED IS DESIRED.

      IF(ISLOP-1) 13,13,14
13    Q1 = (U(2)-U(1))/(X(2)-X(1))
      QN = (U(NPTS)-U(NPTS-1))/(X(NPTS)-X(NPTS-1))
      GO TO 15
14    XL = X(1)
      XR = X(NPTS)
      Q1 = D(XL)
      QN = D(XR)
15    CALL SPLN2
      SUM = 0.0
      M = NPTS-1
      DO 12   I =1,M
      J = I+1
      SUMH = DEL(J)*(U(I)+4.*SPLIT(J)+U(J))/6.
      IF(IPRNT-1) 12,9,12
9     XARG =(I+J)/2.
      WRITE (3,100) XARG,SUMH
100   FORMAT(12X,'PANEL SUM OF THE TOTAL',/,F7.1,E11.3)
 12   SUM = SUM + SUMH
      RETURN
      END

FEATURES SUPPORTED
 NONPROCESS
```

+An example of such a dummy routine is seen in the text, p. 18.

*Copies of these routines may be found in Appendix B.

ACKNOWLEDGEMENTS

The author gratefully acknowledges many discussions with R.F. Thompson and the assistance of T.P. Carleton in optimizing the algorithm.

REFERENCES

Holladay, J.C., "A Smoothest Curve Approximation". Math Tables Aids Comp. 11, (1957).

Thompson, R.F., "Spline Interpolation on a Digital Computer", NASA-GSFC X-692-70-261, 1970.

Thompson, R.F., (Stability Analysis of Cubical Spline) to be published (1971).

APPENDIX A

NPTS	INTEG	SIMPSON $\frac{1}{3}$	EXACT	
5	0.826834E 02	0.826834E 02	0.826834E 02	
7	0.826834E 02	0.826834E 02	0.826834E 02	
9	0.826834E 02	0.826834E 02	0.826834E 02	
11	0.826834E 02	0.826834E 02	0.826834E 02	
13	0.826834E 02	0.826834E 02	0.826834E 02	
15	0.826834E 02	0.826834E 02	0.826834E 02	
17	0.826834E 02	0.826834E 02	0.826834E 02	
19	0.826834E 02	0.826834E 02	0.826834E 02	
21	0.826834E 02	0.826834E 02	0.826834E 02	$\int_0^{2\pi} x^2 dx$
23	0.826834E 02	0.826834E 02	0.826834E 02	
25	0.826834E 02	0.826834E 02	0.826834E 02	
27	0.826834E 02	0.826834E 02	0.826834E 02	
29	0.826834E 02	0.826834E 02	0.826834E 02	
31	0.826834E 02	0.826834E 02	0.826834E 02	
33	0.826834E 02	0.826834E 02	0.826834E 02	
35	0.826834E 02	0.826834E 02	0.826834E 02	
37	0.826834E 02	0.826834E 02	0.826834E 02	
5	0.389636E 03	0.389636E 03	0.389636E 03	
7	0.389636E 03	0.389636E 03	0.389636E 03	
9	0.389636E 03	0.389636E 03	0.389636E 03	
11	0.389636E 03	0.389636E 03	0.389636E 03	
13	0.389636E 03	0.389636E 03	0.389636E 03	
15	0.389636E 03	0.389636E 03	0.389636E 03	
17	0.389636E 03	0.389636E 03	0.389636E 03	
19	0.389636E 03	0.389636E 03	0.389636E 03	$\int_0^{2\pi} x^3 dx$
21	0.389636E 03	0.389636E 03	0.389636E 03	
23	0.389636E 03	0.389636E 03	0.389636E 03	
25	0.389636E 03	0.389636E 03	0.389636E 03	
27	0.389636E 03	0.389636E 03	0.389636E 03	
29	0.389636E 03	0.389636E 03	0.389636E 03	
31	0.389636E 03	0.389636E 03	0.389636E 03	
33	0.389636E 03	0.389636E 03	0.389636E 03	
35	0.389636E 03	0.389636E 03	0.389636E 03	
37	0.389636E 03	0.389636E 03	0.389636E 03	
5	0.195725E 04	0.196362E 04	0.195852E 04	
7	0.195827E 04	0.195953E 04	0.195852E 04	
9	0.195844E 04	0.195884E 04	0.195852E 04	
11	0.195849E 04	0.195865E 04	0.195852E 04	
13	0.195851E 04	0.195858E 04	0.195852E 04	
15	0.195851E 04	0.195855E 04	0.195852E 04	
17	0.195852E 04	0.195854E 04	0.195852E 04	
19	0.195852E 04	0.195853E 04	0.195852E 04	$\int_0^{2\pi} x^4 dx$
21	0.195852E 04	0.195853E 04	0.195852E 04	
23	0.195852E 04	0.195853E 04	0.195852E 04	
25	0.195852E 04	0.195852E 04	0.195852E 04	
27	0.195852E 04	0.195852E 04	0.195852E 04	
29	0.195852E 04	0.195852E 04	0.195852E 04	
31	0.195852E 04	0.195852E 04	0.195852E 04	
33	0.195852E 04	0.195852E 04	0.195852E 04	
35	0.195852E 04	0.195852E 04	0.195852E 04	
37	0.195852E 04	0.195852E 04	0.195852E 04	

NPTS	INTEG	SIMPSON 1/3	EXACT
5	0.102347E 05	0.103349E 05	0.102548E 05
7	0.102508E 05	0.102706E 05	0.102548E 05
9	0.102535E 05	0.102598E 05	0.102548E 05
11	0.102543E 05	0.102568E 05	0.102548E 05
13	0.102545E 05	0.102558E 05	0.102548E 05
15	0.102546E 05	0.102553E 05	0.102548E 05
17	0.102547E 05	0.102551E 05	0.102548E 05
19	0.102547E 05	0.102550E 05	0.102548E 05
21	0.102547E 05	0.102549E 05	0.102548E 05
23	0.102547E 05	0.102549E 05	0.102548E 05
25	0.102548E 05	0.102548E 05	0.102548E 05
27	0.102548E 05	0.102548E 05	0.102548E 05
29	0.102548E 05	0.102548E 05	0.102548E 05
31	0.102548E 05	0.102548E 05	0.102548E 05
33	0.102548E 05	0.102548E 05	0.102548E 05
35	0.102548E 05	0.102548E 05	0.102548E 05
37	0.102548E 05	0.102548E 05	0.102548E 05

$\int_0^{2\pi} x^5 dx$

5	0.549787E 05	0.561900E 05	0.552282E 05
7	0.551786E 05	0.554231E 05	0.552282E 05
9	0.552125E 05	0.552904E 05	0.552282E 05
11	0.552217E 05	0.552538E 05	0.552282E 05
13	0.552251E 05	0.552405E 05	0.552282E 05
15	0.552265E 05	0.552349E 05	0.552282E 05
17	0.552272E 05	0.552321E 05	0.552282E 05
19	0.552276E 05	0.552306E 05	0.552282E 05
21	0.552278E 05	0.552298E 05	0.552282E 05
23	0.552279E 05	0.552293E 05	0.552282E 05
25	0.552280E 05	0.552289E 05	0.552282E 05
27	0.552280E 05	0.552287E 05	0.552282E 05
29	0.552281E 05	0.552286E 05	0.552282E 05
31	0.552281E 05	0.552285E 05	0.552282E 05
33	0.552281E 05	0.552284E 05	0.552282E 05
35	0.552281E 05	0.552284E 05	0.552282E 05
37	0.552281E 05	0.552283E 05	0.552282E 05

$\int_0^{2\pi} x^6 dx$

ƆRD

5	0.300914E 06	0.313714E 06	0.303632E 06
7	0.303090E 06	0.305732E 06	0.303632E 06
9	0.303460E 06	0.304309E 06	0.303632E 06
11	0.303562E 06	0.303912E 06	0.303632E 06
13	0.303598E 06	0.303768E 06	0.303632E 06
15	0.303614E 06	0.303706E 06	0.303632E 06
17	0.303622E 06	0.303675E 06	0.303632E 06
19	0.303626E 06	0.303659E 06	0.303632E 06
21	0.303628E 06	0.303650E 06	0.303632E 06
23	0.303629E 06	0.303645E 06	0.303632E 06
25	0.303630E 06	0.303641E 06	0.303632E 06
27	0.303631E 06	0.303639E 06	0.303632E 06
29	0.303631E 06	0.303637E 06	0.303632E 06
31	0.303632E 06	0.303636E 06	0.303632E 06
33	0.303632E 06	0.303635E 06	0.303632E 06
35	0.303632E 06	0.303635E 06	0.303632E 06
37	0.303632E 06	0.303634E 06	0.303632E 06

$\int_0^{2\pi} x^7 dx$

5	0.166880E 07	0.1791[illegible]E 07	0.169580E 07
7	0.169038E 07	0.171638E 07	0.169580E 07
9	0.169408E 07	0.170250E 07	0.169580E 07
11	0.169509E 07	0.169858E 07	0.169580E 07
13	0.169546E 07	0.169715E 07	0.169580E 07
15	0.169562E 07	0.169653E 07	0.169580E 07
17	0.169569E 07	0.169623E 07	0.169580E 07
19	0.169573E 07	0.169607E 07	0.169580E 07
21	0.169576E 07	0.169598E 07	0.169580E 07
23	0.169577E 07	0.169592E 07	0.169580E 07
25	0.169578E 07	0.169589E 07	0.169580E 07
27	0.169579E 07	0.169586E 07	0.169580E 07
29	0.169579E 07	0.169585E 07	0.169580E 07
31	0.169579E 07	0.169584E 07	0.169580E 07
33	0.169579E 07	0.169583E 07	0.169580E 07
35	0.169580E 07	0.169582E 07	0.169580E 07
37	0.169580E 07	0.169582E 07	0.169580E 07

$\int_0^{2\pi} x^8 dx$

NPTS	INTEG	SIMPSON 1/3	EXACT
5	0.933883E 07	0.104227E 08	0.958956E 07
7	0.953878E 07	0.977690E 07	0.958956E 07
9	0.957335E 07	0.965152E 07	0.958956E 07
11	0.958289E 07	0.961546E 07	0.958956E 07
13	0.958634E 07	0.960219E 07	0.958956E 07
15	0.958781E 07	0.959642E 07	0.958956E 07
17	0.958853E 07	0.959360E 07	0.958956E 07
19	0.958892E 07	0.959208E 07	0.958956E 07
21	0.958914E 07	0.959122E 07	0.958956E 07
23	0.958927E 07	0.959069E 07	0.958956E 07
25	0.958935E 07	0.959036E 07	0.958956E 07
27	0.958941E 07	0.959014E 07	0.958956E 07
29	0.958945E 07	0.958999E 07	0.958956E 07
31	0.958947E 07	0.958989E 07	0.958956E 07
33	0.958949E 07	0.958981E 07	0.958956E 07
35	0.958951E 07	0.958976E 07	0.958956E 07
37	0.958952E 07	0.958971E 07	0.958956E 07

$$\int_0^{2\pi} x^2\,dx$$

CEF

5	0.525643E 08	0.616192E 08	0.547754E 08
7	0.543232E 08	0.563916E 08	0.547754E 08
9	0.546306E 08	0.553193E 08	0.547754E 08
11	0.547157E 08	0.550046E 08	0.547754E 08
13	0.547465E 08	0.548876E 08	0.547754E 08
15	0.547598E 08	0.548365E 08	0.547754E 08
17	0.547662E 08	0.548115E 08	0.547754E 08
19	0.547697E 08	0.547980E 08	0.547754E 08
21	0.547716E 08	0.547903E 08	0.547754E 08
23	0.547728E 08	0.547856E 08	0.547754E 08
25	0.547736E 08	0.547826E 08	0.547754E 08
27	0.547741E 08	0.547806E 08	0.547754E 08
29	0.547744E 08	0.547793E 08	0.547754E 08
31	0.547747E 08	0.547783E 08	0.547754E 08
33	0.547748E 08	0.547777E 08	0.547754E 08
35	0.547750E 08	0.547772E 08	0.547754E 08
37	0.547750E 08	0.547768E 08	0.547754E 08

$$\int_0^{2\pi} x^{10}\,dx$$

5	0.715231669E 00	0.715244137E 00
7	0.715233658E 00	0.715236111E 00
9	0.715233993E 00	0.715234767E 00
11	0.715234084E 00	0.715234402E 00
13	0.715234117E 00	0.715234270E 00
15	0.715234130E 00	0.715234213E 00
17	0.715234136E 00	0.715234186E 00
19	0.715234139E 00	0.715234170E 00
21	0.715234142E 00	0.715234162E 00
23	0.715234144E 00	0.715234157E 00
25	0.715234143E 00	0.715234154E 00
27	0.715234144E 00	0.715234150E 00
29	0.715234143E 00	0.715234149E 00
31	0.715234143E 00	0.715234149E 00
33	0.715234144E 00	0.715234148E 00
35	0.715234143E 00	0.715234147E 00
37	0.715234144E 00	0.715234146E 00

EXACT

0.715234148E 00

$$\int_0^{\pi/4} \operatorname{sech} x\,dx$$

NPTS	SPLINE	SIMPSON 1/3	
5	0.292892613F 00	0.292895648E 00	
7	0.292893099F 00	0.292893697E 00	
9	0.292893180F 00	0.292893369E 00	
11	0.292893203E 00	0.292893280E 00	
13	0.292893210E 00	0.292893248F 00	
15	0.292893214F 00	0.292893234E 00	
17	0.292893215E 00	0.292893227F 00	
19	0.292893216E 00	0.292893224E 00	
21	0.292893217E 00	0.292893222E 00	0.292893219E 00
23	0.292893217F 00	0.292893220E 00	
25	0.292893217E 00	0.292893220E 00	
27	0.292893217E 00	0.292893219E 00	
29	0.292893217F 00	0.292893218F 00	$\int_0^{\pi/4} \sin x \, dx$
31	0.292893217F 00	0.292893219E 00	
33	0.292893217F 00	0.292893218F 00	
35	0.292893217E 00	0.292893218E 00	
37	0.292893217E 00	0.292893218F 00	

NPTS	SPLINE	SIMPSON 1/3	
5	0.157160684E-07	0.146291807E-07	
7	0.215368345E-07	0.213748586E-07	
9	-0.183354131E-08	-0.792413960E-08	
11	0.366708263F-08	0.112157052E-08	
13	0.196159817F-07	0.185709322E-07	
15	0.424915925F-08	0.247302818E-08	
17	-0.416184775E-08	-0.512021327E-08	
19	-0.171712599E-08	0.647476705E-08	0.0
21	0.288855517E-08	-0.853368880E-09	
23	0.820728019E-08	-0.746974837E-08	
25	0.226827978F-08	-0.692854256E-08	$\int_0^{\pi/4} \cos(100x) \, dx$
27	-0.982936399E-09	0.395738096E-08	
29	0.135514710E-09	0.440616992E-08	
31	0.411091605F-09	0.307212796E-08	
33	-0.600266503E-09	0.472400629E-09	
35	-0.261934474E-08	-0.573693364E-09	
37	0.978616299F-09	0.327802014E-08	

NPTS	SPLINE	SIMPSON 1/3	
5	0.214443513E 00	0.215150730E 00	
7	0.214569839E 00	0.214721053E 00	
9	0.214591627E 00	0.214640998E 00	
11	0.214597637E 00	0.214618174E 00	
13	0.214599806F 00	0.214609796E 00	
15	0.214600739E 00	0.214606159E 00	
17	0.214601192E 00	0.214604381E 00	
19	0.214601434F 00	0.214603429E 00	
21	0.214601572E 00	0.214602883E 00	0.214601834E 00
23	0.214601655F 00	0.214602552E 00	
25	0.214601708E 00	0.214602342E 00	
27	0.214601743E 00	0.214602204E 00	
29	0.214601766E 00	0.214602109E 00	
31	0.214601783E 00	0.214602043E 00	$\int_0^{\pi/4} \tan^2 x \, dx$
33	0.214601795F 00	0.214601996E 00	
35	0.214601804E 00	0.214601962E 00	
37	0.214601810E 00	0.214601936E 00	

NPTS	SPLINE	SIMPSON 1/3	
5	0.119327758E 01	0.119328985E 01	
7	0.119327956E 01	0.119328199E 01	
9	0.119327989E 01	0.119328066E 01	
11	0.119327998E 01	0.119328030E 01	
13	0.119328001F 01	0.119328017E 01	
15	0.119328003E 01	0.119328011E 01	
17	0.119328003F 01	0.119328008E 01	
19	0.119328004E 01	0.119328007E 01	
21	0.119328004F 01	0.119328006E 01	0.119328004E 01
23	0.119328004F 01	0.119328005E 01	
25	0.119328004E 01	0.119328005E 01	
27	0.119328004F 01	0.119328005E 01	
29	0.119328004E 01	0.119328005E 01	
31	0.119328004E 01	0.119328004E 01	$\int_0^{\pi/4} \exp(x) \, dx$
33	0.119328004F 01	0.119328004F 01	
35	0.119328004F 01	0.119328004E 01	
37	0.119328004F 01	0.119328004E 01	

NPTS	SPLINE	SIMPSON 1/3	EXACT	
5	0.126487651E 04	0.135282391E 04	0.128696609E 04	
7	0.126228003E 04	0.126382290E 04	0.126259857E 04	
9	0.126249733E 04	0.126299443E 04	0.126259857E 04	
11	0.126255701E 04	0.126276236E 04	0.126259857E 04	
13	0.126257850E 04	0.126267800E 04	0.126259857E 04	
15	0.126258773E 04	0.126264158E 04	0.126259857E 04	
17	0.126259221E 04	0.126262384E 04	0.126259857E 04	
19	0.126259459E 04	0.126261437E 04	0.126259857E 04	
21	0.126259596E 04	0.126260894E 04	0.126259857E 04	$\int_{\pi/16}^{5\pi/2} \sinh x\,dx$
23	0.126259678E 04	0.126260566E 04	0.126259857E 04	
25	0.126259730E 04	0.126260357E 04	0.126259857E 04	
27	0.126259765E 04	0.126260220E 04	0.126259857E 04	
29	0.126259788E 04	0.126260127E 04	0.126259857E 04	
31	0.126259805E 04	0.126260062E 04	0.126259857E 04	
33	0.126259816E 04	0.126260015E 04	0.126259857E 04	
35	0.126259825E 04	0.126259981E 04	0.126259857E 04	
37	0.126259831E 04	0.126259955E 04	0.126259857E 04	
5	0.1[illegible]02531E 04	0.126845534E 04	0.126261788E 04	
7	0.126229933E 04	0.126384223E 04	0.126261788E 04	
9	0.126251664E 04	0.126301375E 04	0.126261788E 04	
11	0.126257632E 04	0.126278168E 04	0.126261788E 04	
13	0.126259782E 04	0.126269731E 04	0.126261788E 04	
15	0.126260704E 04	0.126266089E 04	0.126261788E 04	
17	0.126261152E 04	0.126264315E 04	0.126261788E 04	
19	0.126261391E 04	0.126263368E 04	0.126261788E 04	
21	0.126261527E 04	0.126262825E 04	0.126261788E 04	
23	0.126261610E 04	0.126262497E 04	0.126261788E 04	$\int_{\pi/16}^{5\pi/2} \cosh x\,dx$
25	0.126261662E 04	0.126262289E 04	0.126261788E 04	
27	0.126261697E 04	0.126262152E 04	0.126261788E 04	
29	0.126261720E 04	0.126262058E 04	0.126261788E 04	
31	0.126261736E 04	0.126261993E 04	0.126261788E 04	
33	0.126261748E 04	0.126261946E 04	0.126261788E 04	
35	0.126261756E 04	0.126261912E 04	0.126261788E 04	
37	0.126261763E 04	0.126261887E 04	0.126261788E 04	
5	0.392661015E 01	0.392658126E 01	0.392660283E 01	
7	0.392660435E 01	0.392659761E 01	0.392660283E 01	
9	0.392660332E 01	0.392660104E 01	0.392660283E 01	
11	0.392660303E 01	0.392660207E 01	0.392660283E 01	
13	0.392660293E 01	0.392660245E 01	0.392660283E 01	
15	0.392660288E 01	0.392660262E 01	0.392660283E 01	
17	0.392660285E 01	0.392660271E 01	0.392660283E 01	
19	0.392660284E 01	0.392660275E 01	0.392660283E 01	
21	0.392660283E 01	0.392660277E 01	0.392660283E 01	$\int_{\pi/16}^{5\pi/2} \tanh x\,dx$
23	0.392660283E 01	0.392660279E 01	0.392660283E 01	
25	0.392660282E 01	0.392660279E 01	0.392660283E 01	
27	0.392660282E 01	0.392660280E 01	0.392660283E 01	
29	0.392660283E 01	0.392660280E 01	0.392660283E 01	
31	0.392660282E 01	0.392660280E 01	0.392660283E 01	
33	0.392660282E 01	0.392660281E 01	0.392660283E 01	
35	0.392660281E 01	0.392660281E 01	0.392660283E 01	
37	0.392660281E 01	0.392660281E 01	0.392660283E 01	
5	0.392737157E 01	0.392740057E 01	0.392737894E 01	
7	0.392737741E 01	0.392738418E 01	0.392737894E 01	
9	0.392737844E 01	0.392738073E 01	0.392737894E 01	
11	0.392737873E 01	0.392737969E 01	0.392737894E 01	
13	0.392737883E 01	0.392737931E 01	0.392737894E 01	
15	0.392737887E 01	0.392737913E 01	0.392737894E 01	
17	0.392737889E 01	0.392737905E 01	0.392737894E 01	
19	0.392737890E 01	0.392737900E 01	0.392737894E 01	
21	0.392737891E 01	0.392737898E 01	0.392737894E 01	
23	0.392737891E 01	0.392737896E 01	0.392737894E 01	$\int_{\pi/16}^{5\pi/2} \mathrm{ctnh}\, x\,dx$
25	0.392737891E 01	0.392737895E 01	0.392737894E 01	
27	0.392737892E 01	0.392737894E 01	0.392737894E 01	
29	0.392737891E 01	0.392737893E 01	0.392737894E 01	
31	0.392737891E 01	0.392737893E 01	0.392737894E 01	
33	0.392737891E 01	0.392737893E 01	0.392737894E 01	
35	0.392737891E 01	0.392737892E 01	0.392737894E 01	
37	0.392737892E 01	0.392737892E 01	0.392737894E 01	

APPENDIX B

```
      SUBROUTINE SPLN2
      DIMENSION A(100),V(100)
      COMMON X(100),U(100),S(100),DEL(100),Q1,QN,NPTS
C
C                 UNEQUALLY SPACED DATA
C                 *********************
C           THIS PROGRAM COMPUTES THE SECOND DERIVATIVE, S(X), OF THE
C           CUBIC SPLINE, SPLIN(X), WHICH INTERPOLATES THE NPTS OF
C           ARBITRARILY SPACED DATA (X,U).
C           THE COMMON STATEMENT PROVIDES COMMUNICATION WITH THE
C           FOLLOWING FUNCTION SUBROUTINES
C
C           FUNCTION
C           SUBROUTINE   DESCRIPTION
C
C           SPLIN(X)     FOR THE VALUE X, SPLIN RETURNS THE VALUE OF THE
C                        CUBIC SPLINE WHICH INTERPOLATES THE NPTS
C                        DATA POINTS (X,U)
C           DSPLN(X)     RETURNS THE DERIVATIVE OF THE SPLINE AT X.
C
C
      N=NPTS
      IF(N-3) 5,5,1
    1 IF(N-100) 6,6,5
C
C           COMPUTE DEL AND V
C
    6 DEL(2)=X(2)-X(1)
      V(1)=6.0*(((U(2)-U(1))/DEL(2))-Q1 )
      N1=N-1
      DO 2 I=2,N1
      DEL(I+1)=X(I+1)-X(I)
    2 V(I)=((U(I-1)/DEL(I))-U(I)*((1.0/DEL(I))+(1.0/DEL(I+1)))
     C    +(U(I+1)/DEL(I+1)))*6.0
      V(N)=(QN+(U(N1)-U(N))/DEL(N) )*6.0
C
C           GAUSSIAN ELIMINATION AND AUGMENTATION
C
      A(1)=2.0*DEL(2)
      A(2)=1.5*DEL(2)+2.0*DEL(3)
      V(2)=V(2)-0.5*V(1)
      DO 3 I=3,N1
      C=DEL(I)/A(I-1)
      A(I)=2.0*(DEL(I)+DEL(I+1))-C*DEL(I)
      V(I)=V(I)-C*V(I-1)
    3 CONTINUE
      C=DEL(N)/A(N1)
      A(N)=2.0*DEL(N)-C*DEL(N)
      V(N)=V(N)-C*V(N1)
C
C           BACK SUBSTITUTION
      S(N)=V(N)/A(N)
      DO 4 J=1,N1
      I=N-J
    4 S(I)=(V(I)-DEL(I+1)*S(I+1))/A(I)
    5 RETURN
      END
```

APPENDIX B

```
      SUBROUTINE SPLN2
      DIMENSION A(100),V(100)
      COMMON X(100),U(100),S(100),DEL(100),Q1,QN,NPTS
C
C                 UNEQUALLY SPACED DATA
C                 *********************
C           THIS PROGRAM COMPUTES THE SECOND DERIVATIVE, S(X), OF THE
C           CUBIC SPLINE, SPLIN(X), WHICH INTERPOLATES THE NPTS OF
C           ARBITRARILY SPACED DATA (X,U).
C           THE COMMON STATEMENT PROVIDES COMMUNICATION WITH THE
C           FOLLOWING FUNCTION SUBROUTINES
C
C           FUNCTION
C           SUBROUTINE    DESCRIPTION
C
C           SPLIN(X)      FOR THE VALUE X, SPLIN RETURNS THE VALUE OF THE
C                         CUBIC SPLINE WHICH INTERPOLATES THE NPTS
C                         DATA POINTS (X,U)
C           DSPLN(X)      RETURNS THE DERIVATIVE OF THE SPLINE AT X.
C
C
      N=NPTS
      IF(N-3) 5,5,1
    1 IF(N-100) 6,6,5
C
C           COMPUTE DEL AND V
C
    6 DEL(2)=X(2)-X(1)
      V(1)=6.0*(((U(2)-U(1))/DEL(2))-Q1 )
      N1=N-1
      DO 2 I=2,N1
      DEL(I+1)=X(I+1)-X(I)
    2 V(I)=((U(I-1)/DEL(I))-U(I)*((1.0/DEL(I))+(1.0/DEL(I+1)))
     C     +(U(I+1)/DEL(I+1)))*6.0
      V(N)=(QN+(U(N1)-U(N))/DEL(N) )*6.0
C
C           GAUSSIAN ELIMINATION AND AUGMENTATION
C
      A(1)=2.0*DEL(2)
      A(2)=1.5*DEL(2)+2.0*DEL(3)
      V(2)=V(2)-0.5*V(1)
      DO 3 I=3,N1
      C=DEL(I)/A(I-1)
      A(I)=2.0*(DEL(I)+DEL(I+1))-C*DEL(I)
      V(I)=V(I)-C*V(I-1)
    3 CONTINUE
      C=DEL(N)/A(N1)
      A(N)=2.0*DEL(N)-C*DEL(N)
      V(N)=V(N)-C*V(N1)
C
C           BACK SUBSTITUTION
      S(N)=V(N)/A(N)
      DO 4 J=1,N1
      I=N-J
    4 S(I)=(V(I)-DEL(I+1)*S(I+1))/A(I)
    5 RETURN
      END
```

```
      FUNCTION SPLIT(N)

                THIS SUBROUTINE IS MEANT TO BE USED IN CONJUNCTION WITH
                SPLN2 AND INTEG AND RETURNS THE VALUE OF THE CUBICAL
                SPLINE INTERPOLANT AT THE MIDPOINT OF THE N'TH INTERVAL
                BETWEEN DATA POINTS.

      COMMON X(100),U(100),S(100),DEL(100),Q1,QN,NPTS
      D = DEL(N)/2.0
      SPLIT = (U(N)+U(N-1))/2.0-D*D*(S(N)+S(N-1))/4.0
      RETURN
      END
```

FEATURES SUPPORTED